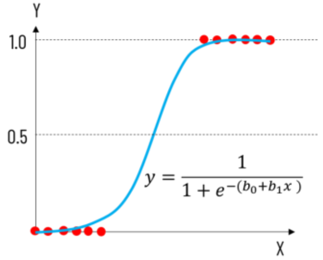
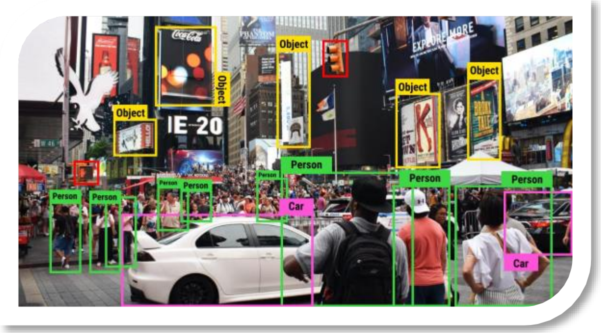
COMPUTATIONAL
CHEMISTRY
DAY 2023

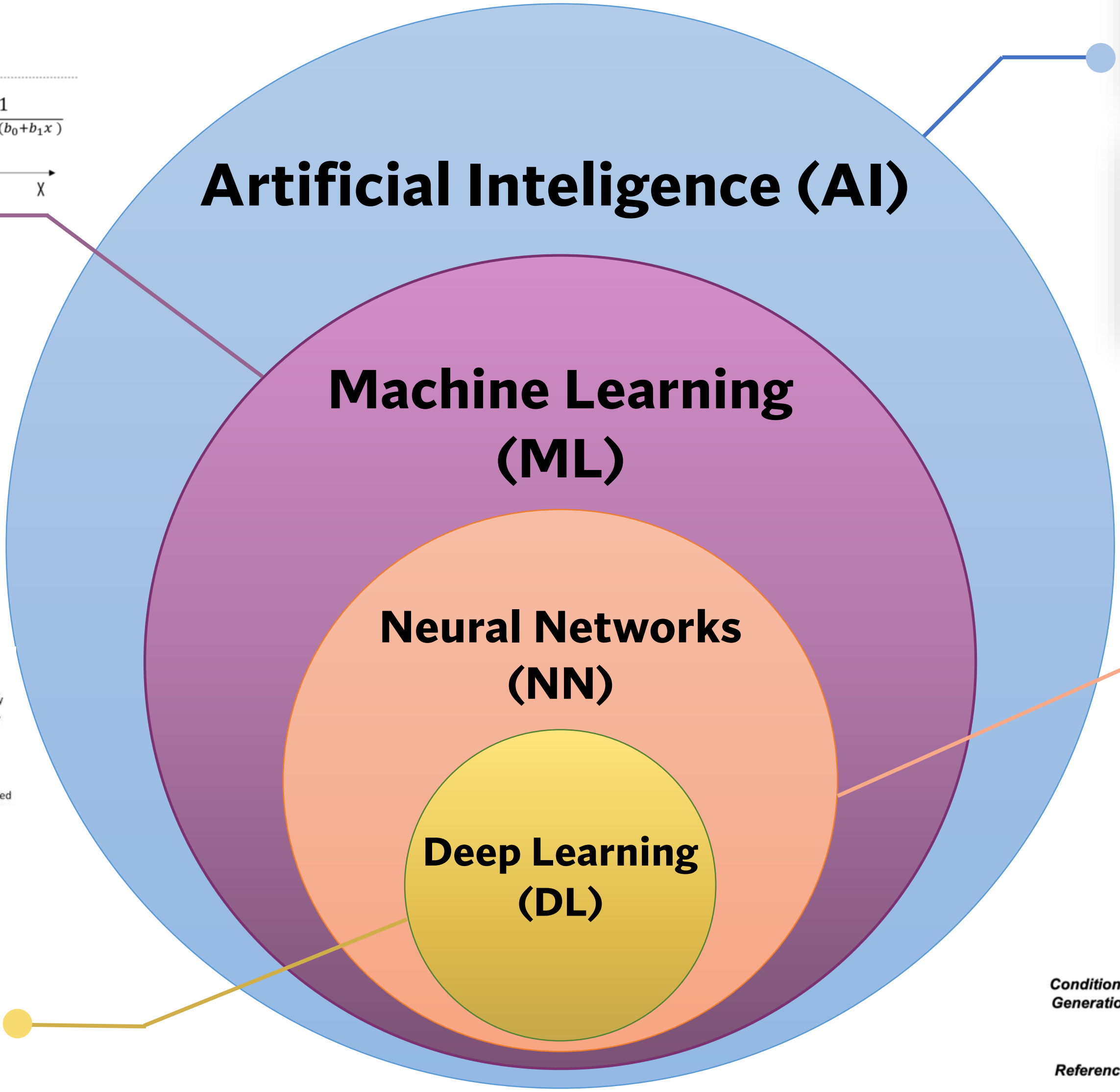
Modelling Multidimensional PESs by Deep Reinforcement Learning

Dr. Karlo Sović
Department of Chemistry
University of Zagreb Faculty of Science

COMPUTATIONAL CHEMISTRY DAY 2023





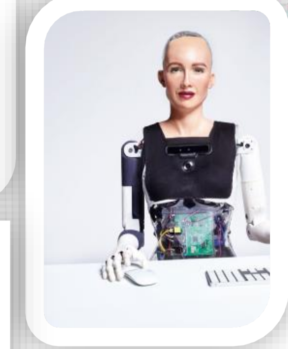


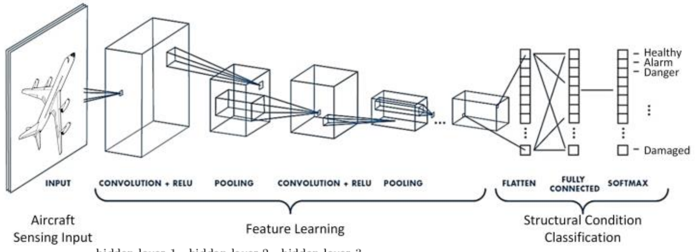
Artificial Inteligence (AI)

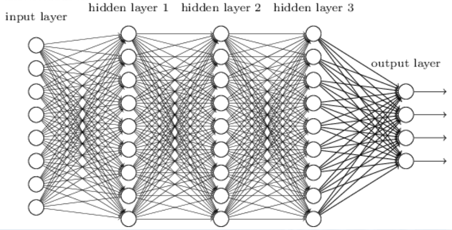
Machine Learning (ML)

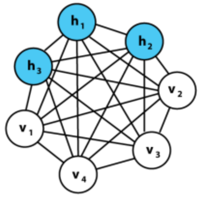
Neural Networks (NN)

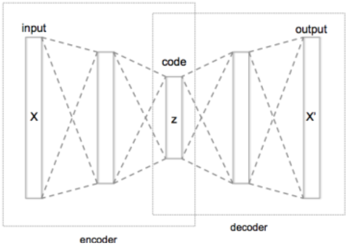
Deep Learning (DL)

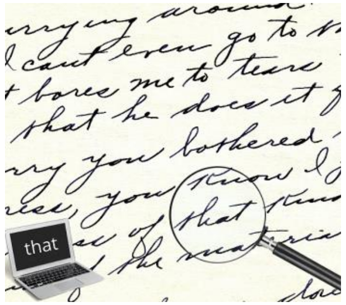




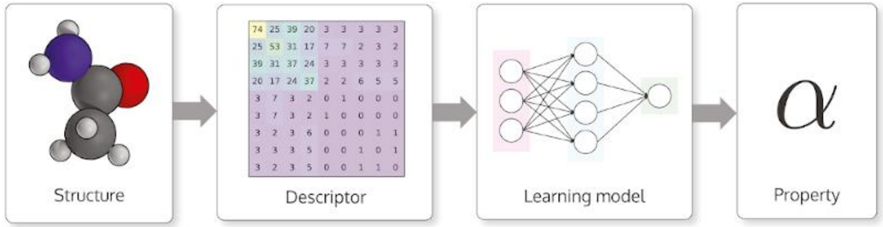




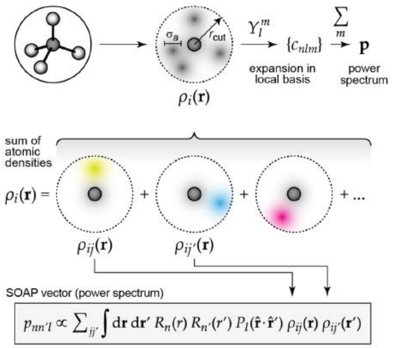




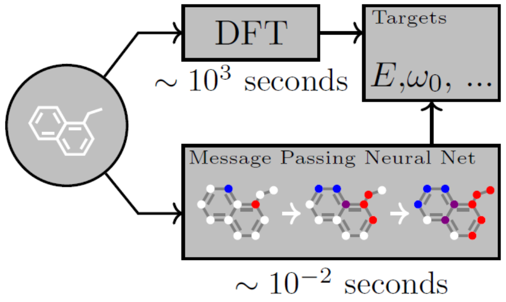
Machine and deep learning methods



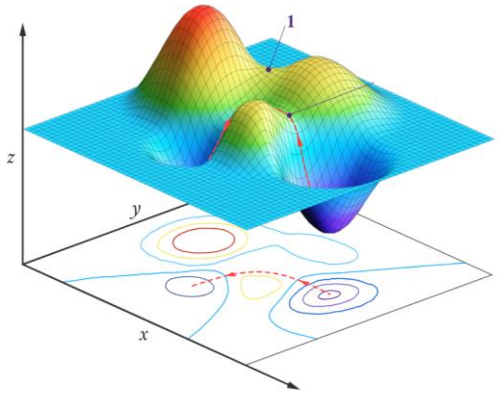
Predicting molecular properties



Modelling electron and charge density



Neural message passing



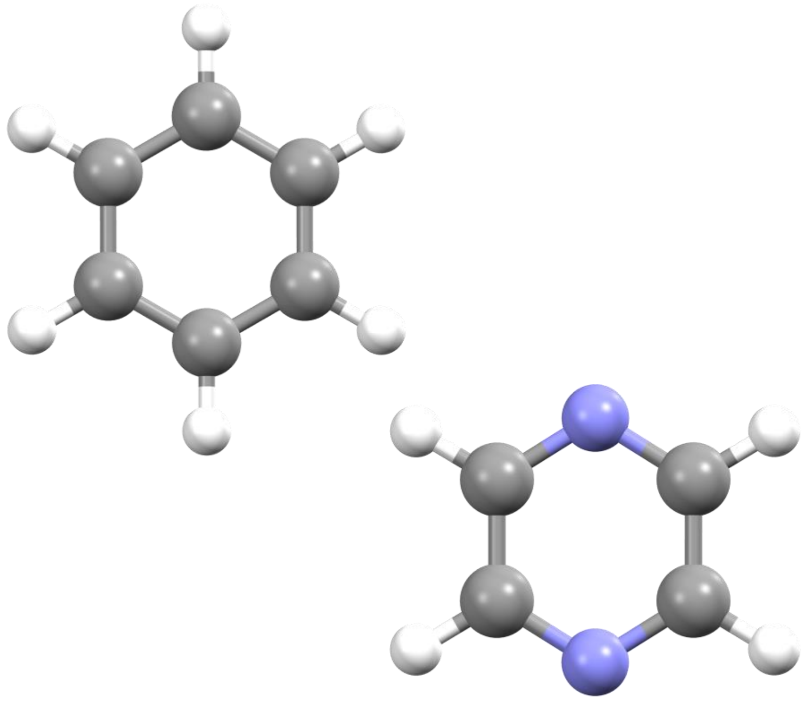
Modelling PES

Content

- Modelling *n*-D PES cuts of benzene and pyrazine with respect to their normal coordinates

1) Using machine-learning MLR algorithm.

2) Using feed-forward multi-layer neural networks utilizing *supervised* and *deep reinforcement learning*.



Multiple linear regression (MLR)

Multiple linear regression

- *Multiple linear regression* (MLR) model has the form

$$y_i = b_0 + \sum_{j=1}^{p} b_j x_{ij} + e_i$$

for $i \in \{1, \ldots, n\}$ where

$y_i \in \mathbb{R}$ is the real-valued response for the i-th observation
$b_0 \in \mathbb{R}$ is the regression intercept
$b_j \in \mathbb{R}$ is the j-th predictor's regression slope
$x_{ij} \in \mathbb{R}$ is the j-th predictor for the i-th observation
$e_i \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian error term.

MLR model in matrix notation

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e}$$

$\boldsymbol{y} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ is the $n \times 1$ response vector

$\boldsymbol{X} = \left[\mathbf{1}_n, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_p\right] \in \mathbb{R}^{n \times (p+1)}$ is the $n \times (p+1)$ **design matrix**

$\boldsymbol{b} = \left(b_0, b_1, \ldots, b_p\right)^T \in \mathbb{R}^{p+1}$ is the $(p+1) \times 1$ vector of coefficients

$\boldsymbol{e} = (e_1, \ldots, e_n)^T \in \mathbb{R}^n$ is the $n \times 1$ error vector

Multiple linear regression (MLR)

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ 1 & x_{31} & x_{32} & \cdots & x_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{pmatrix}$$

n – number of samples

p – number of independent variables

Multiple linear regression (MLR)

- Fitted values are given by

$$\widehat{\boldsymbol{y}} = \boldsymbol{X}\widehat{\boldsymbol{b}}$$

and the residuals are given by

$$\hat{\boldsymbol{e}} = \boldsymbol{y} - \widehat{\boldsymbol{y}}$$

- The ordinary least squares (OLS) problem is

$$\min_{\boldsymbol{b}\in\mathbb{R}^{p+1}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|^2 = \min_{\boldsymbol{b}\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} \left(y_i - b_0 - \sum\nolimits_{j=1}^{p} b_j x_{ij} \right)^2$$

- The OLS solution has the form

$$\boxed{\widehat{\boldsymbol{b}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}}$$

Some of the algorithms for solving this equation include: **SVD**, QR decomposition, LU decomposition, Givens transformations, Cholesky decomposition, etc.

Multiple linear regression (MLR)

Relevant sum-of-squares

- In MLR models, the relevant *sum-of-squares* are

Sum-of-squares **T**otal $\mathrm{SST} = \sum_{i=1}^{n} (y_i - \bar{y})^2$

Sum-of-squares **R**egression $\mathrm{SSR} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$

Sum-of-squares **E**rror $\mathrm{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

- The corresponding degrees of freedom are

SST: $df_T = n - 1$

SSR: $df_R = p$

SSE: $df_E = n - p - 1$

The coefficient of multiple determination is defined as

$$R^2 = \frac{\mathrm{SSR}}{\mathrm{SST}} = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}}$$

- Including more predictors in MLR model can artificially inflate R^2

The adjusted R^2 is a relative measure of fit

$$R^2_{\mathrm{adj.}} = 1 - \frac{\mathrm{SSE}/df_E}{\mathrm{SST}/df_T}$$

$R^2_{\mathrm{pred.}}$ → average $R^2_{\mathrm{adj.}}$ from LOO-CV

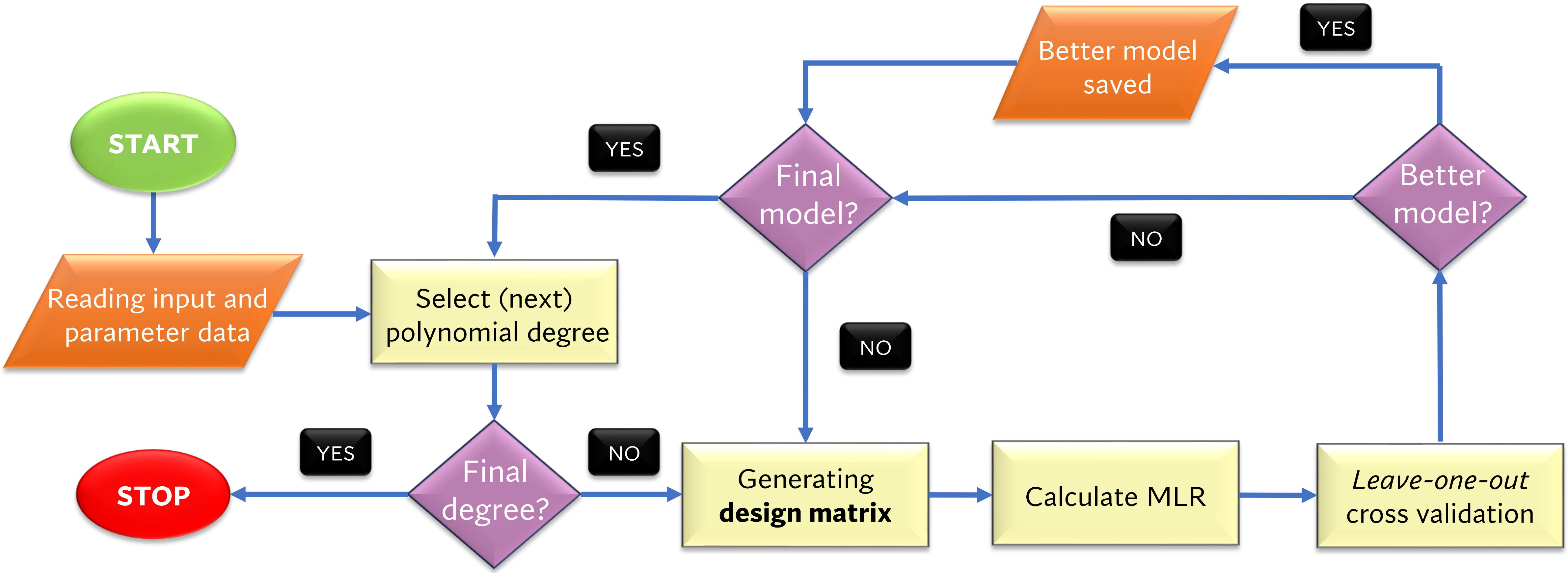
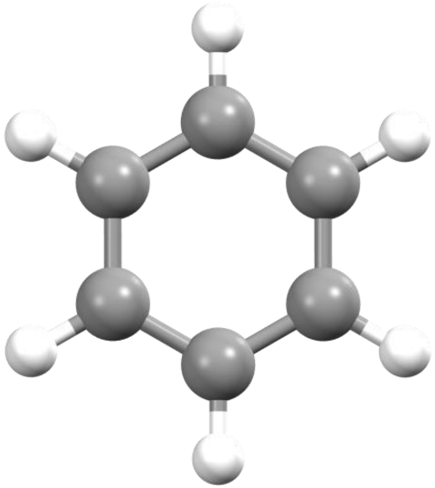

START
Reading input and parameter data
Select (next) polynomial degree
Final degree?
YES
STOP
NO
Generating **design matrix**
Calculate MLR
Leave-one-out cross validation
Better model?
YES
Better model saved
NO
Final model?
YES
NO

Figure 1. Flowchart for machine-learning MLR algorithm implemented in ***moonee*** program.

Multiple linear regression (MLR) ▸ Results

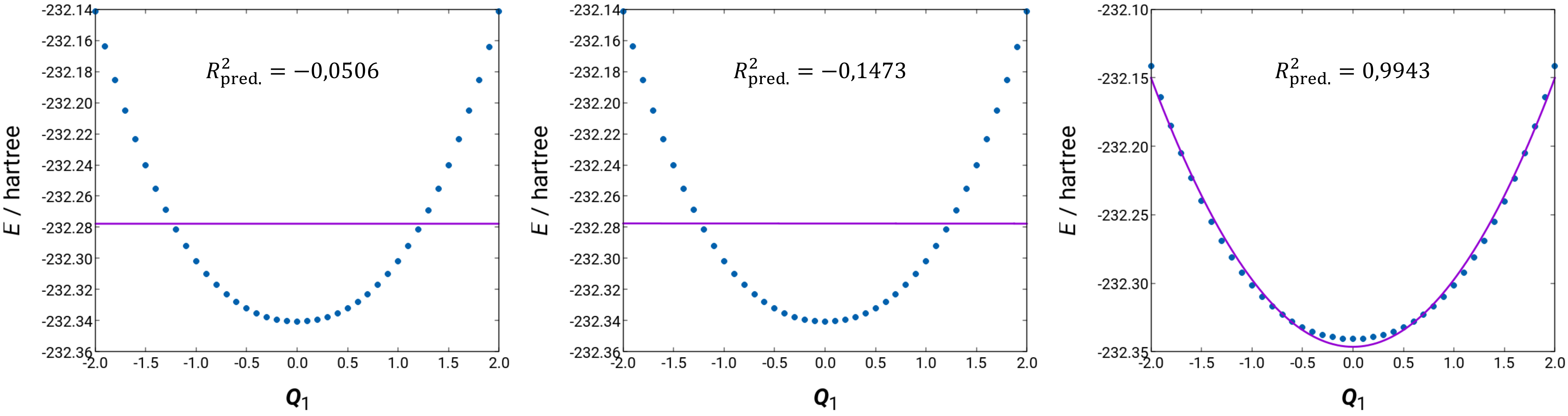


- A total of 64 models were calculated for each 1D cut of benzene’s PES.
- A total of 1920 models were calculated for **all** 1D cuts of benzene’s PES.

Table 1. Calculated regression models of 1D cut of benzene’s PES with respect to its 1st normal coordinate using ***moonee*** ML-MLR algorithm.

Degree of polynomial	$\mathbf{R^2_{pred.}}$	**MSE**	**Regression model**
0	−0,0506	$3{,}34 \cdot 10^{-3}$	$-2{,}32 \cdot 10^{2}$
1	−0,1473	$3{,}34 \cdot 10^{-3}$	$-3{,}51 \cdot 10^{-5}\, \boldsymbol{Q}_1 - 2{,}32 \cdot 10^{2}$
2	0,9943	$1{,}72 \cdot 10^{-5}$	$4{,}91 \cdot 10^{-2}\, \boldsymbol{Q}_1^{2} - 2{,}32 \cdot 10^{2}$
3	0,9936	$1{,}71 \cdot 10^{-5}$	$-1{,}18 \cdot 10^{-5}\, \boldsymbol{Q}_1^{3} + 4{,}91 \cdot 10^{-2}\, \boldsymbol{Q}_1^{2} - 2{,}32 \cdot 10^{2}$
4	0,9998	$4{,}81 \cdot 10^{-7}$	$3{,}22 \cdot 10^{-3}\, \boldsymbol{Q}_1^{4} + 3{,}75 \cdot 10^{-2}\, \boldsymbol{Q}_1^{2} - 2{,}32 \cdot 10^{2}$
5	0,9997	$4{,}87 \cdot 10^{-7}$	$-3{,}18 \cdot 10^{-6}\, \boldsymbol{Q}_1^{5} + 3{,}22 \cdot 10^{-3}\, \boldsymbol{Q}_1^{4} + 3{,}76 \cdot 10^{-2}\, \boldsymbol{Q}_1^{2} - 2{,}32 \cdot 10^{2}$
6	0,9999	$5{,}20 \cdot 10^{-9}$	$-5{,}25 \cdot 10^{-4}\, \boldsymbol{Q}_1^{6} + 6{,}21 \cdot 10^{-3}\, \boldsymbol{Q}_1^{4} + 3{,}34 \cdot 10^{-2}\, \boldsymbol{Q}_1^{2} - 3{,}51 \cdot 10^{-5}\, \boldsymbol{Q}_1 - 2{,}32 \cdot 10^{2}$

Multiple linear regression (MLR) ▸ Results



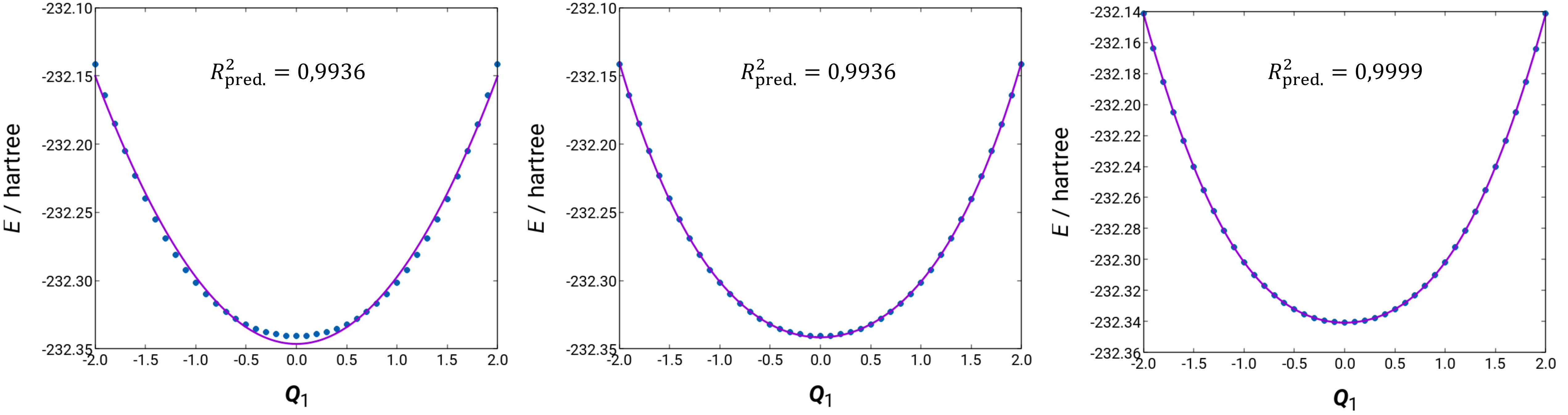
$E = -2{,}32 \cdot 10^2$

$E = -3{,}51 \cdot 10^{-5}\, \mathbf{Q}_1 - 2{,}32 \cdot 10^2$

$E = 4{,}91 \cdot 10^{-2}\, \mathbf{Q}_1^{\,2} - 2{,}32 \cdot 10^2$

Figure 2. Regreesion models of 1D cut of benzene's PES with respect to its 1st normal coordinate ($\mathbf{Q}_1$) for 0th degree polynomial (left), 1st degree polynomial (center) and 2nd degree polynomial (right) calculated using machine-learning MLR algorithm implemented in program ***moonee***.

Multiple linear regression (MLR) ▸ Results



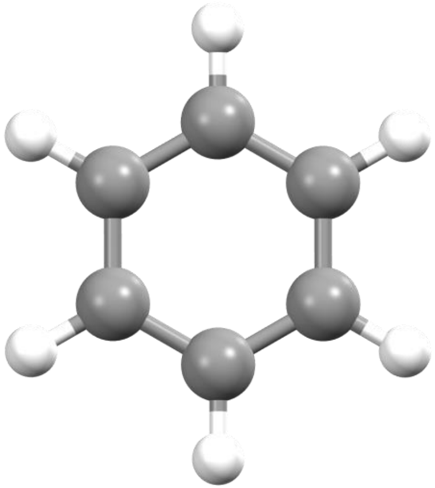
$$E = -1{,}18 \cdot 10^{-5}\, \mathbf{Q}_1^{3} + 4{,}91 \cdot 10^{-2}\, \mathbf{Q}_1^{2} - 2{,}32 \cdot 10^{2}$$

$$E = 3{,}22 \cdot 10^{-3}\, \mathbf{Q}_1^{4} + 3{,}75 \cdot 10^{-2}\, \mathbf{Q}_1^{2} - 2{,}32 \cdot 10^{2}$$

$$E = -5{,}25 \cdot 10^{-4}\, \mathbf{Q}_1^{6} + 6{,}21 \cdot 10^{-3} \mathbf{Q}_1^{4} + 3{,}34 \cdot 10^{-2}\, \mathbf{Q}_1^{2} - 3{,}51 \cdot 10^{-5}\, \mathbf{Q}_1 - 2{,}32 \cdot 10^{2}$$

Figure 3. Regreesion models of 1D cut of benzene's PES with respect to its 1st normal coordinate ($\mathbf{Q}_1$) for 3rd degree polynomial (left), 4th degree polynomial (center) and 6th degree polynomial (right) calculated using machine-learning MLR algorithm implemented in program ***moonee***.

Multiple linear regression (MLR) ▸ Results



- For all 1D cross sections:

$$\overline{R^2}_{\text{pred.}} = 0{,}9992$$

$$\overline{\text{MSE}} = 8{,}91 \cdot 10^{-7}$$

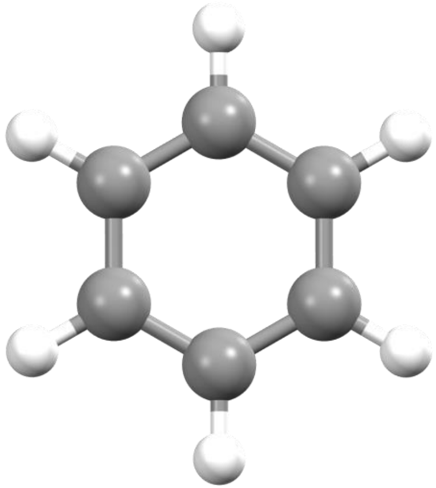
Table 2. Depicted regression models of 1D cuts of benzene's PES with respect to its normal coordinates calculated using ***moonee*** ML-MLR algorithm.

Normal coordinate	$R^2_{\text{pred.}}$	MSE	Regression model
1	0,9998	$4{,}81 \cdot 10^{-7}$	$3{,}22 \cdot 10^{-3}\, \mathbf{Q}_1^4 + 3{,}75 \cdot 10^{-2}\, \mathbf{Q}_1^2 - 2{,}32 \cdot 10^2$
2	0,9998	$4{,}82 \cdot 10^{-7}$	$3{,}16 \cdot 10^{-3}\, \mathbf{Q}_2^4 + 3{,}76 \cdot 10^{-2}\, \mathbf{Q}_2^2 - 2{,}32 \cdot 10^2$
3	0,9989	$8{,}21 \cdot 10^{-6}$	$1{,}83 \cdot 10^{-1}\, \mathbf{Q}_3^2 + 1{,}83 \cdot 10^{-3}\, \mathbf{Q}_3 - 2{,}32 \cdot 10^2$
4	0,9986	$7{,}70 \cdot 10^{-6}$	$1{,}83 \cdot 10^{-1}\, \mathbf{Q}_4^2 + 3{,}14 \cdot 10^{-3}\, \mathbf{Q}_4 - 2{,}32 \cdot 10^2$
5	0,9994	$1{,}85 \cdot 10^{-6}$	$8{,}63 \cdot 10^{-3}\, \mathbf{Q}_5^4 + 4{,}92 \cdot 10^{-2}\, \mathbf{Q}_5^2 - 2{,}32 \cdot 10^2$
6	0,9996	$2{,}59 \cdot 10^{-6}$	$1{,}22 \cdot 10^{-1}\, \mathbf{Q}_6^2 - 2{,}32 \cdot 10^2$
7	0,9996	$7{,}17 \cdot 10^{-7}$	$1{,}30 \cdot 10^{-2}\, \mathbf{Q}_7^4 + 7{,}87 \cdot 10^{-2}\, \mathbf{Q}_7^2 - 2{,}32 \cdot 10^2$
⋮	⋮	⋮	⋮
30	0,9960	$5{,}87 \cdot 10^{-6}$	$-6{,}85 \cdot 10^{-1}\, \mathbf{Q}_{30}^3 + 8{,}55 \cdot 10^{-1}\, \mathbf{Q}_{30}^2 - 2{,}32 \cdot 10^2$

Multiple linear regression (MLR) ▸ Results

Table 3. Depicted regression models of 2D cuts of benzene's PES with respect to its normal coordinates calculated using ***moonee*** ML-MLR algorithm.

Normal coordinates	$R^2_{\text{pred.}}$	MSE	Regression model
3,4	0,9984	$3{,}82 \cdot 10^{-5}$	$1{,}91 \cdot 10^{-1}\, \boldsymbol{Q}_3^{\,2} + 1{,}89 \cdot 10^{-1}\, \boldsymbol{Q}_4^{\,2} + 2{,}48 \cdot 10^{-3}\, \boldsymbol{Q}_3\, \boldsymbol{Q}_4 - 1{,}87 \cdot 10^{-3}\, \boldsymbol{Q}_4 - 2{,}32 \cdot 10^{2}$
3,8	0,9986	$5{,}65 \cdot 10^{-6}$	$1{,}73 \cdot 10^{-1}\, \boldsymbol{Q}_3^{\,2} + 8{,}82 \cdot 10^{-2}\, \boldsymbol{Q}_8^{\,2} - 1{,}22 \cdot 10^{-3}\, \boldsymbol{Q}_3 - 2{,}32 \cdot 10^{2}$
6,17	0,9985	$1{,}52 \cdot 10^{-5}$	$1{,}17 \cdot 10^{-5}\, \boldsymbol{Q}_6^{\,2} + 1{,}10 \cdot 10^{-1}\, \boldsymbol{Q}_{17}^{\,2} - 2{,}40 \cdot 10^{-3}\, \boldsymbol{Q}_{17} - 2{,}32 \cdot 10^{2}$
8,11	0,9999	$7{,}82 \cdot 10^{-7}$	$1{,}89 \cdot 10^{-1}\, \boldsymbol{Q}_8^{\,4} + 7{,}08 \cdot 10^{-2}\, \boldsymbol{Q}_{11}^{\,4} - 6{,}79 \cdot 10^{-3}\, \boldsymbol{Q}_8^{\,2}\boldsymbol{Q}_8^{\,2} - 1{,}13 \cdot 10^{-3}\, \boldsymbol{Q}_8^{\,2}\boldsymbol{Q}_{11} - 2{,}27 \cdot 10^{-1}\, \boldsymbol{Q}_{11}^{\,3} + 7{,}15 \cdot 10^{-2}\, \boldsymbol{Q}_8^{\,2} + 4{,}17 \cdot 10^{-2}\, \boldsymbol{Q}_{11}^{\,2} + 4{,}57 \cdot 10^{-3}\, \boldsymbol{Q}_{11} - 2{,}32 \cdot 10^{2}$
17,20	0,9994	$2{,}58 \cdot 10^{-6}$	$1{,}23 \cdot 10^{-1}\, \boldsymbol{Q}_{17}^{\,2} + 1{,}71 \cdot 10^{-1}\, \boldsymbol{Q}_{20}^{\,2} + 2{,}48 \cdot 10^{-3}\, \boldsymbol{Q}_{17}\boldsymbol{Q}_{20} - 2{,}92 \cdot 10^{-3}\, \boldsymbol{Q}_{17} - 2{,}32 \cdot 10^{2}$
20,25	0,9995	$2{,}24 \cdot 10^{-6}$	$2{,}26 \cdot 10^{-2}\, \boldsymbol{Q}_{20}^{\,4} + 2{,}77 \cdot 10^{-1}\, \boldsymbol{Q}_{25}^{\,4} - 4{,}01 \cdot 10^{-1}\, \boldsymbol{Q}_{20}^{\,2}\boldsymbol{Q}_{25}^{\,2} + 1{,}52 \cdot 10^{-1}\, \boldsymbol{Q}_{20}^{\,2} + 7{,}37 \cdot 10^{-1}\, \boldsymbol{Q}_{25}^{\,2} - 2{,}32 \cdot 10^{2}$

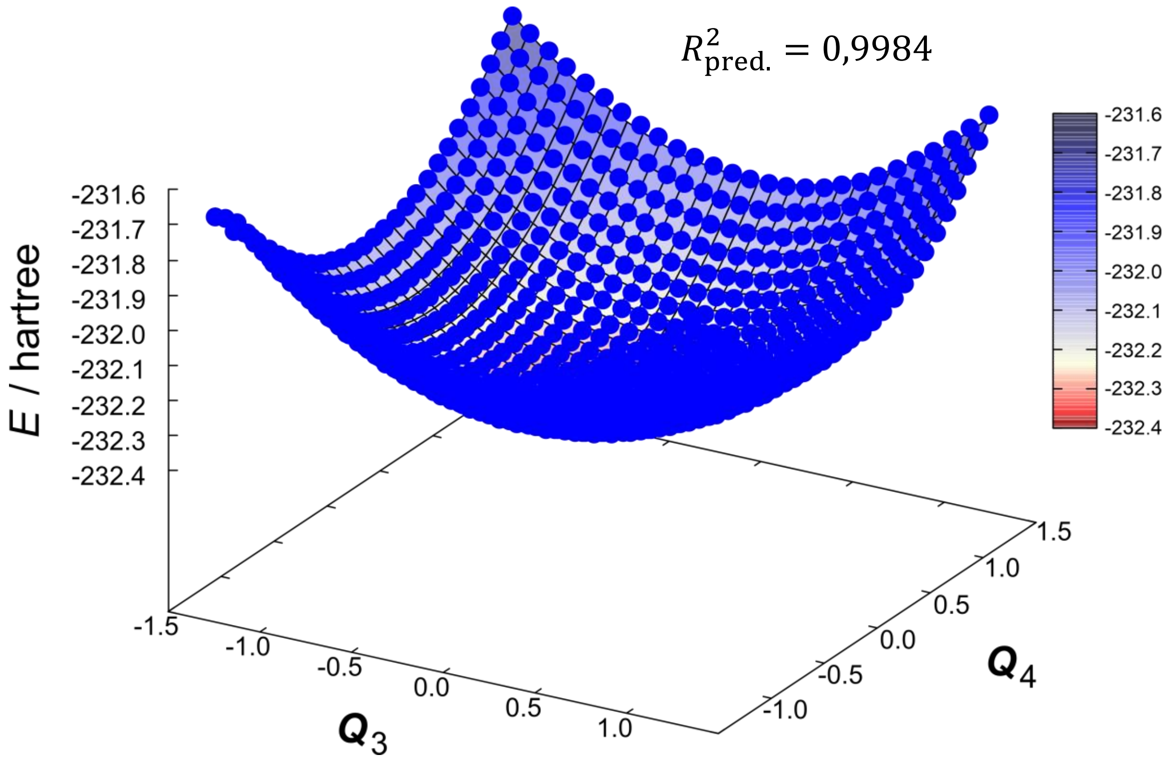


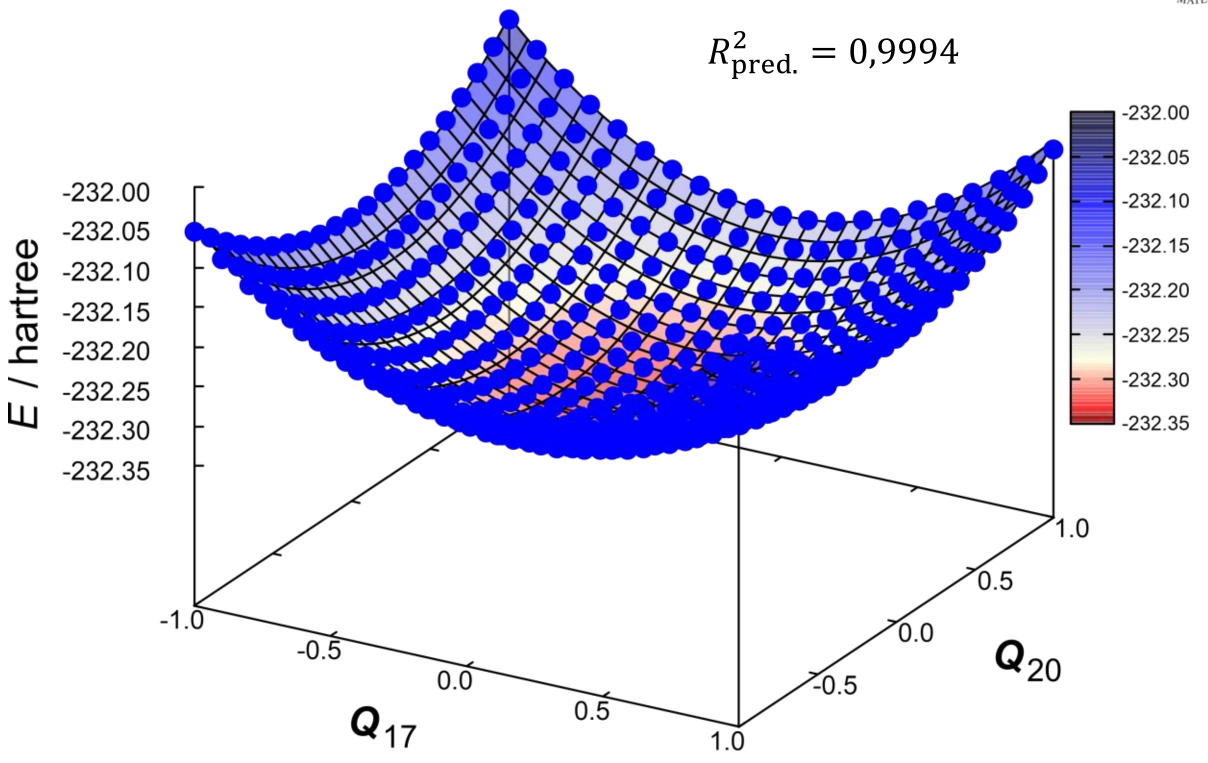
- A total of 16384 models were calculated for each 2D cut of benzene's PES.

$$\overline{R^2}_{\text{pred.}} = 0{,}9985$$

$$\overline{\text{MSE}} = 3{,}52 \cdot 10^{-6}$$

Multiple linear regression (MLR) ▸ Results



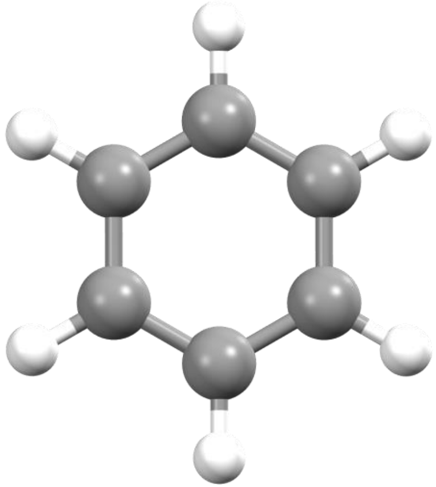


$E = 1{,}91 \cdot 10^{-1}\, \boldsymbol{Q}_3^{\,2} + 1{,}89 \cdot 10^{-1}\, \boldsymbol{Q}_4^{\,2} + 2{,}48 \cdot 10^{-3}\, \boldsymbol{Q}_3\boldsymbol{Q}_4 - 1{,}87 \cdot 10^{-3}\, \boldsymbol{Q}_4 - 2{,}32 \cdot 10^{2}$

$E = 1{,}23 \cdot 10^{-1}\, \boldsymbol{Q}_{17}^{\,2} + 1{,}71 \cdot 10^{-1}\, \boldsymbol{Q}_{20}^{\,2} + 2{,}48 \cdot 10^{-3}\, \boldsymbol{Q}_{17}\boldsymbol{Q}_{20} - 2{,}92 \cdot 10^{-3}\, \boldsymbol{Q}_{17} - 2{,}32 \cdot 10^{2}$

Figure 4. Regression models of 2D cuts of benzene's PES with respect to its 3rd and 4th normal coordinates (left), and to its 17th and 20th normal coordinates (right) calculated using machine-learning MLR algorithm implemented in program ***moonee***.

Multiple linear regression (MLR) ▸ Results



- A total of 524288 models up to 3rd degree of polynomial were calculated for 3D cut of benzene's PES.

- A total of 16384 models up to 2nd degree of polynomial were calculated for 4D cut of benzene's PES.

Table 4. Depicted regression model of 3D and 4D cut of benzene's PES with respect to its normal coordinates calculated using ***moonee*** ML-MLR algorithm.

Normal coordinates	**$R^2_{\mathrm{pred.}}$**	**MSE**	**Regression model**
3,4,7	0,9991	$7{,}69\cdot10^{-6}$	$5{,}31\cdot10^{-3}\,\boldsymbol{Q}_3^{3} + 1{,}25\cdot10^{-2}\,\boldsymbol{Q}_3\boldsymbol{Q}_7^{2} + 2{,}11\cdot10^{-1}\,\boldsymbol{Q}_3^{2}$ $+ 2{,}08\cdot10^{-1}\,\boldsymbol{Q}_4^{2} - 6{,}45\cdot10^{-3}\,\boldsymbol{Q}_3 - 2{,}32\cdot10^{2}$
3,4,7,15	0,9961	$6{,}37\cdot10^{-5}$	$2{,}24\cdot10^{-1}\,\boldsymbol{Q}_3^{2} + 2{,}27\cdot10^{-1}\,\boldsymbol{Q}_4^{2} + 9{,}60\cdot10^{-2}\,\boldsymbol{Q}_7^{2}$ $+ 1{,}31\cdot10^{-1}\,\boldsymbol{Q}_{15}^{2} + 3{,}80\cdot10^{-2}\,\boldsymbol{Q}_3\boldsymbol{Q}_{15}$ $+ 1{,}82\cdot10^{-3}\,\boldsymbol{Q}_3 - 5{,}97\cdot10^{-4}\,\boldsymbol{Q}_{15} - 2{,}32\cdot10^{2}$

Neural networks (NN)

Neural networks

Neural networks (NN) are the quintessential deep learning models that are used to approximate an arbitrary function

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

with high accuracy using brain-like architecture of multiple connections between nodes (*neurons*) which are represented by compositions of activation functions.

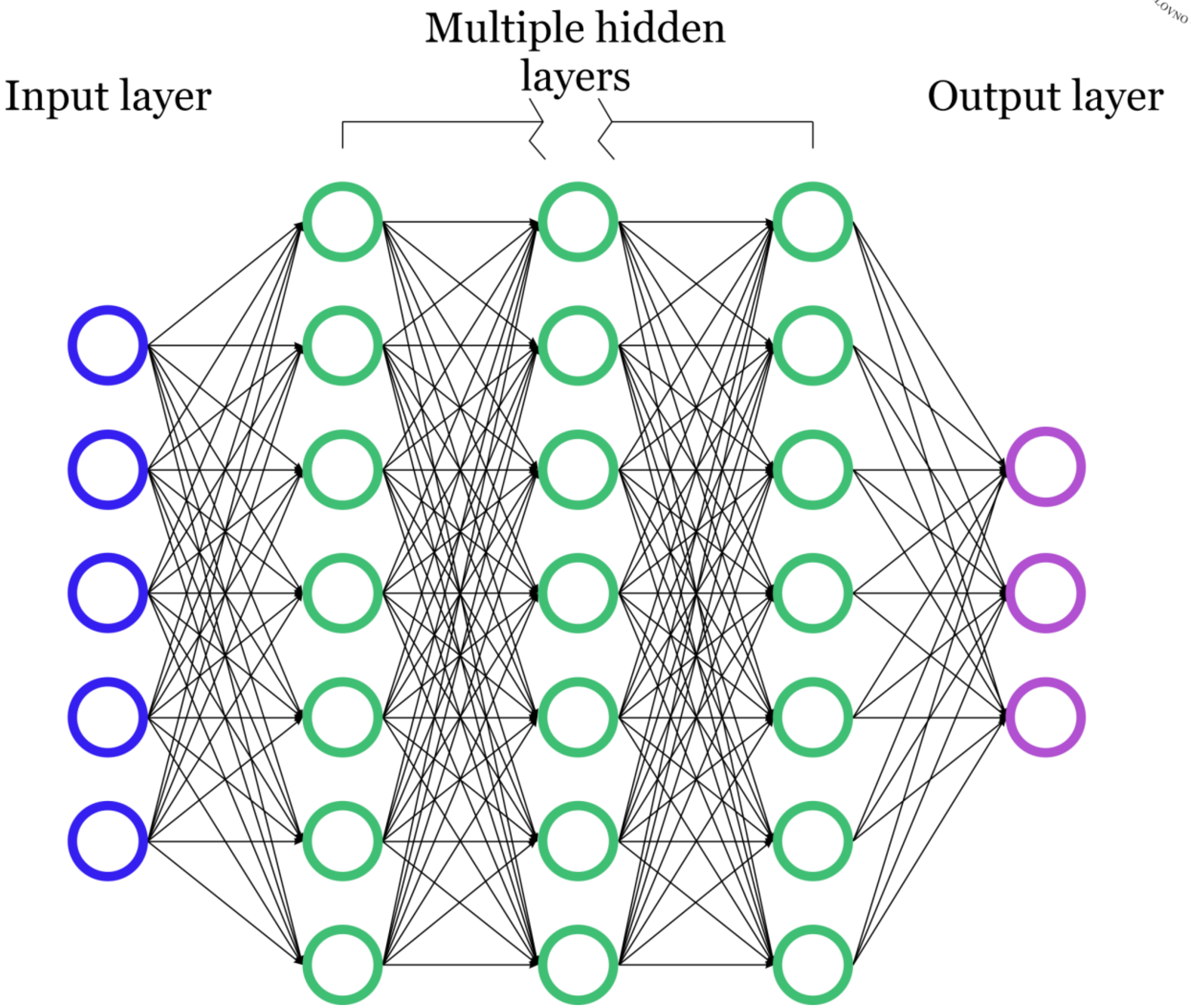
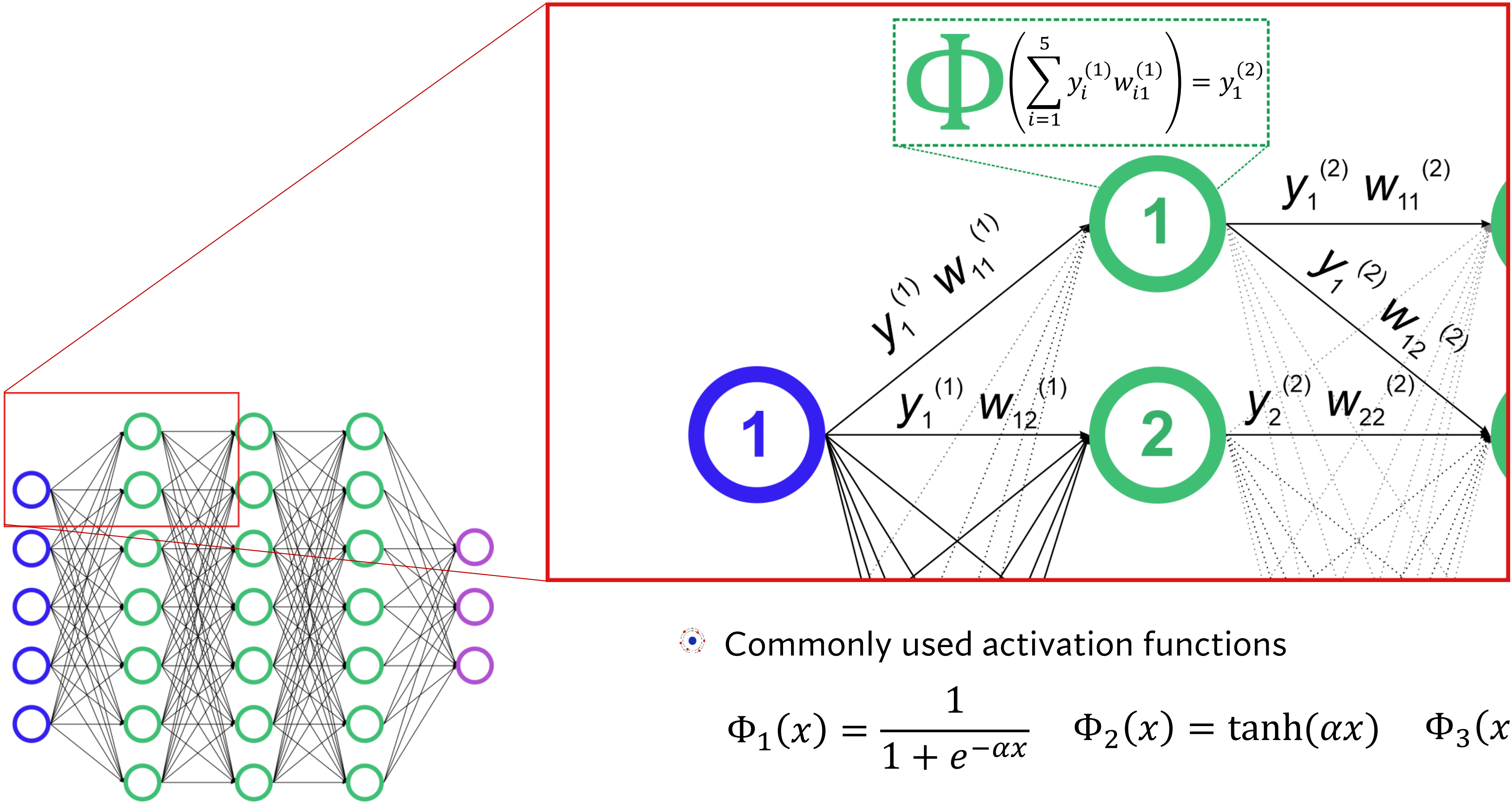


Figure 5. Graphical representation of a 5-layer feed-forward neural network with 5 × 7 × 7 × 7 × 3 architecture.

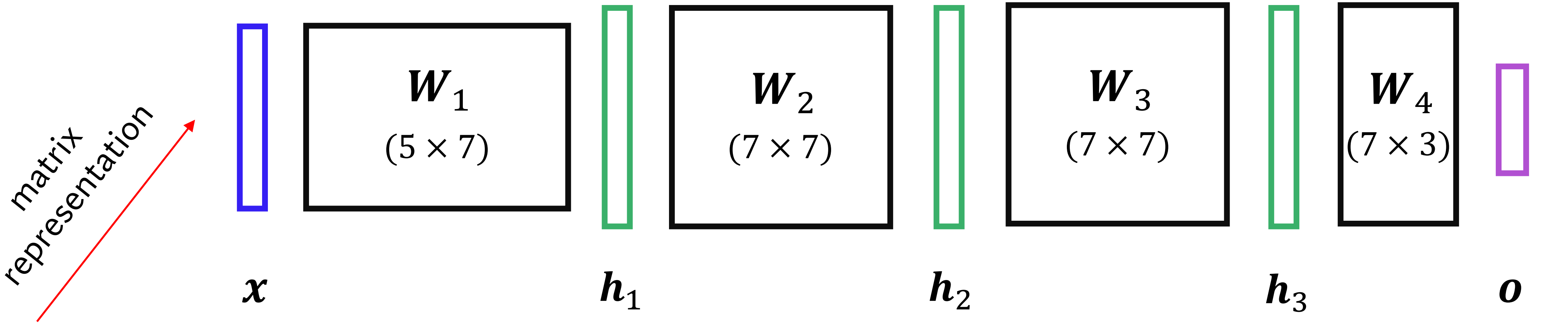
Neural networks (NN)


$\Phi\left(\sum_{i=1}^{5} y_i^{(1)} w_{i1}^{(1)}\right) = y_1^{(2)}$
1
1
2
$y_1^{(1)}\ w_{11}^{(1)}$
$y_1^{(1)}\ w_{12}^{(1)}$
$y_1^{(2)}\ w_{11}^{(2)}$
$y_1^{(2)}\ w_{12}^{(2)}$
$y_2^{(2)}\ w_{22}^{(2)}$

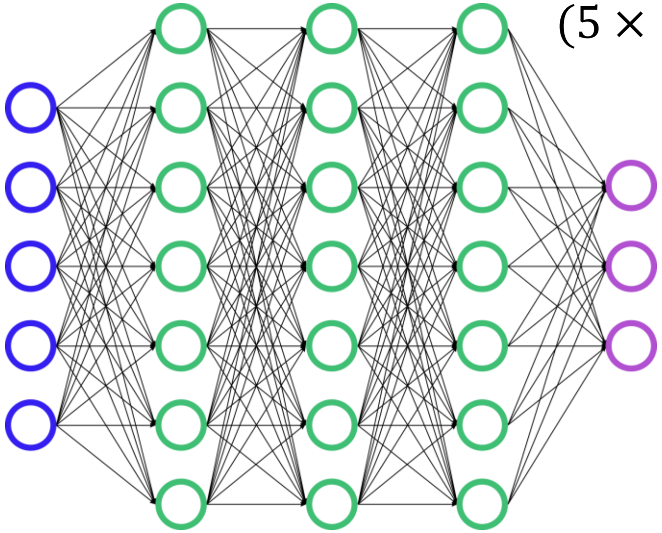
- Commonly used activation functions

$$\Phi_1(x) = \frac{1}{1+e^{-\alpha x}} \quad \Phi_2(x) = \tanh(\alpha x) \quad \Phi_3(x) = \begin{cases} 1, \text{if } x > 0 \\ 0, \text{if } x = 0 \\ -1, \text{if } x < 0 \end{cases}$$

Neural networks (NN)



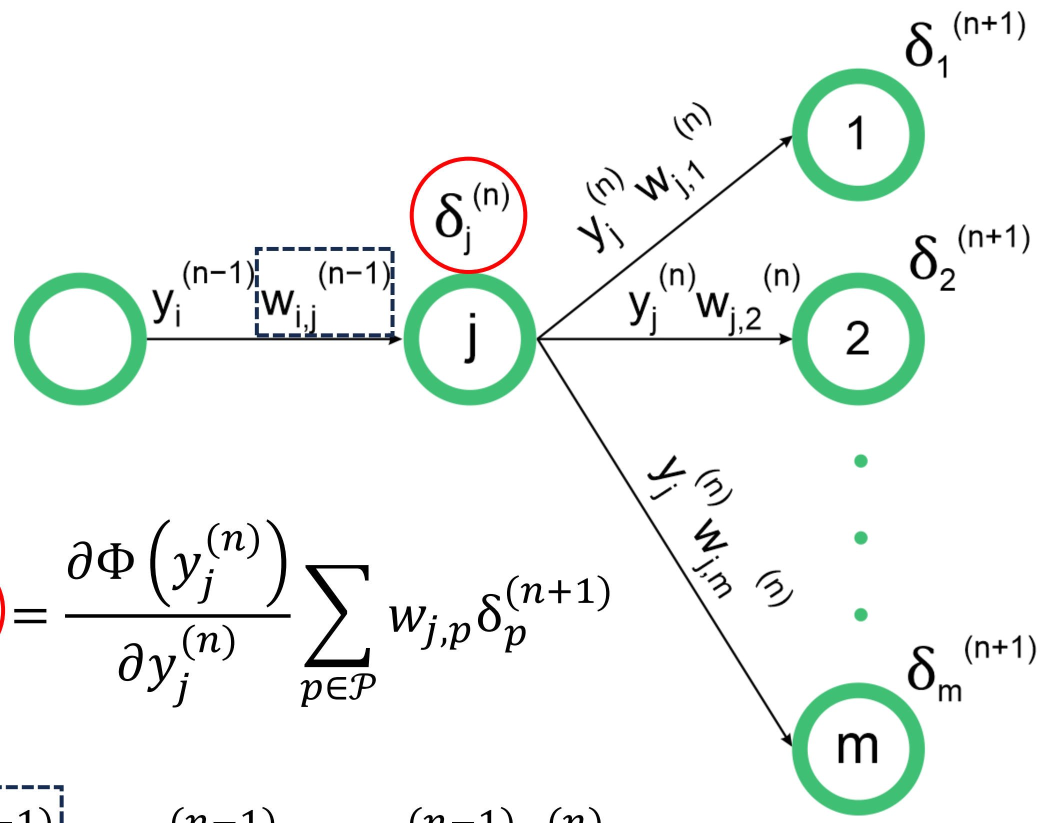
(5×1) (7×1) (7×1) (7×1) (3×1)



$$h_1 = \Phi(W_1^T x) \qquad h_2 = \Phi(W_2^T h_1) \qquad h_3 = \Phi(W_3^T h_2)$$

$$o = \Phi(W_4^T h_3)$$

Neural networks (NN)



$$\boxed{\delta_j^{(n)}} = \frac{\partial \Phi\left(y_j^{(n)}\right)}{\partial y_j^{(n)}} \sum_{p \in \mathcal{P}} w_{j,p} \delta_p^{(n+1)}$$

$$\boxed{w_{i,j}^{(n-1)}} = w_{i,j}^{(n-1)} + \underbrace{\alpha}_{\text{learning rate}} y_i^{(n-1)} \delta_j^{(n)}$$

- The **backpropagation** algorithm first uses a *forward phase* in order to compute the output and the loss function.
- In supervised learning:

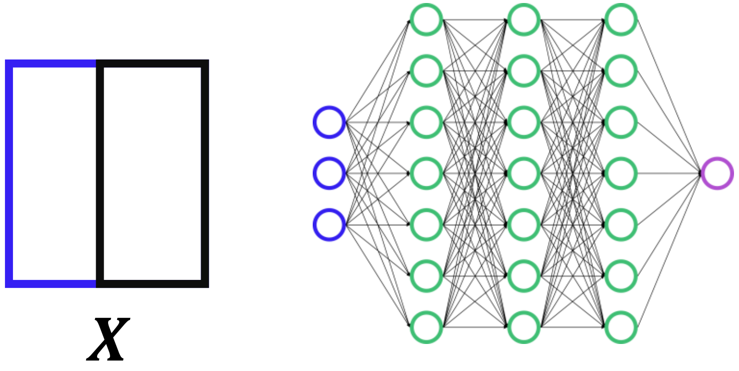
$$(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T \rightarrow (o_1, o_2, \ldots, o_N)^T; \quad (d_1, d_2, \ldots, d_N)^T$$

$$L = \frac{1}{2} \sum_{i=1}^{N} (o_i - d_i)^2$$

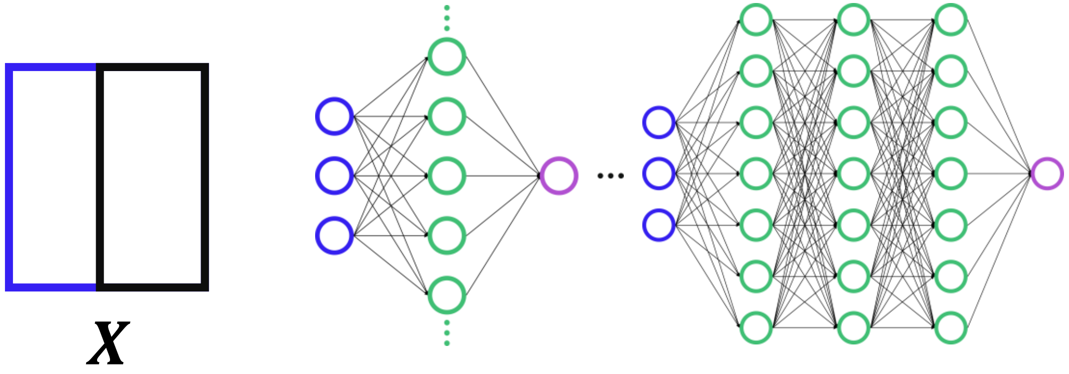
- The *backward phase* computes the gradient of the loss function with respect to various weights. Subsequently, the derivatives are propagated in the backwards direction using the multivariable chain rule.

Neural networks (NN) ▸ Algorithms

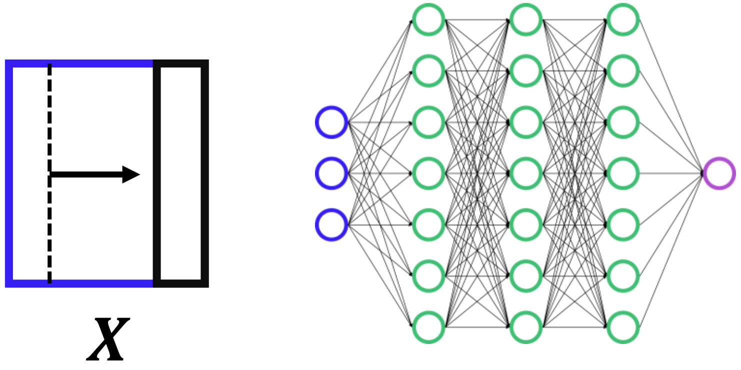
1) *Supervised learning* of neural network with *defined* architecture and data set.



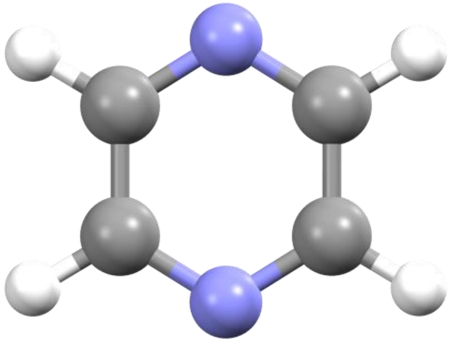
2) *Supervised learning* of neural networks with combinatorial investigation of various architectures for a given data set.

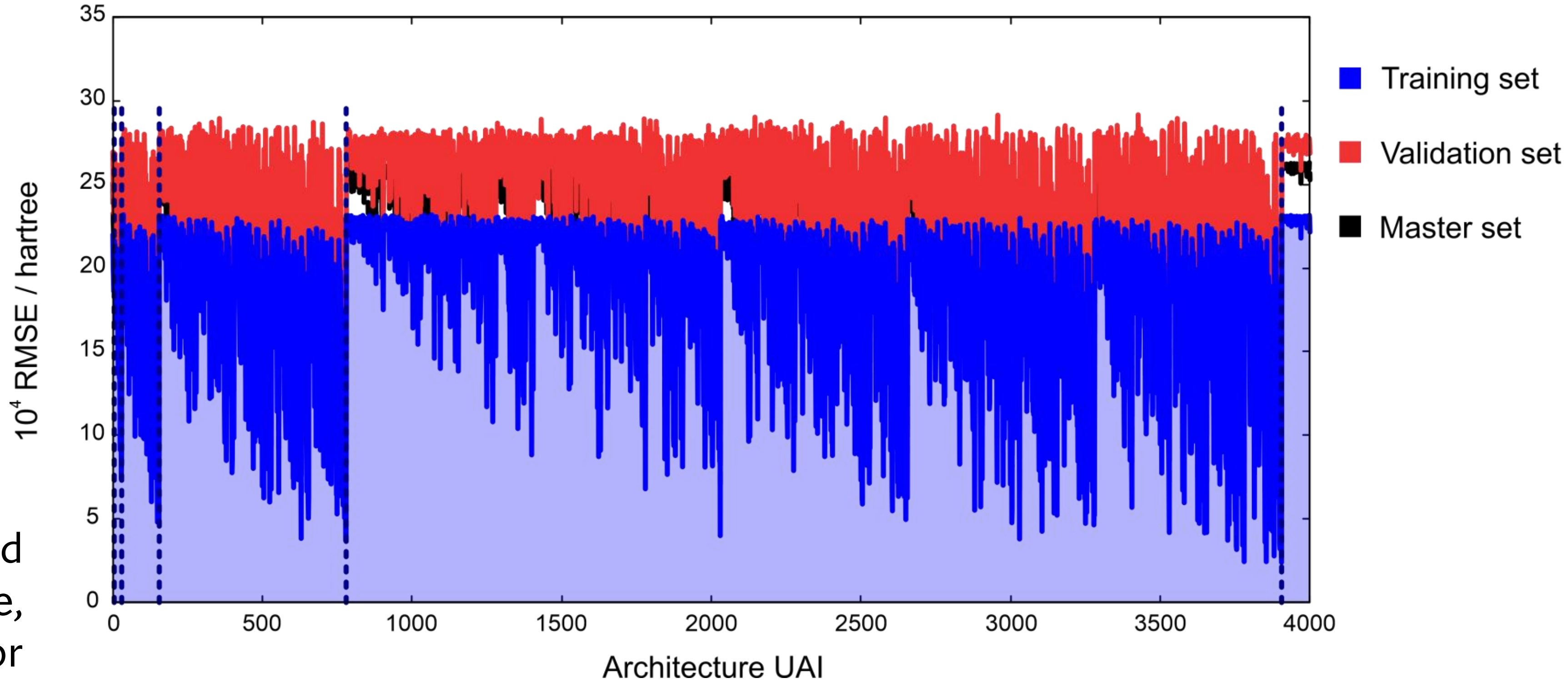


3) *Deep reinforcement learning* of neural networks with *defined* architecture but with progressively increasing training set for a given data set.



Neural networks (NN) ▸ Results





- For a total of 4000 calculated NNs with different architecture, optimal was obtained for $1 \times (400)^8 \times 1$ architecture with RMSE of 3,81 $\cdot 10^{-4}$ hartree.

Figure 6. Neural network architecture analysis for modelling 1D cut of pyrazine's PES with respect to its 1st normal coordinate.

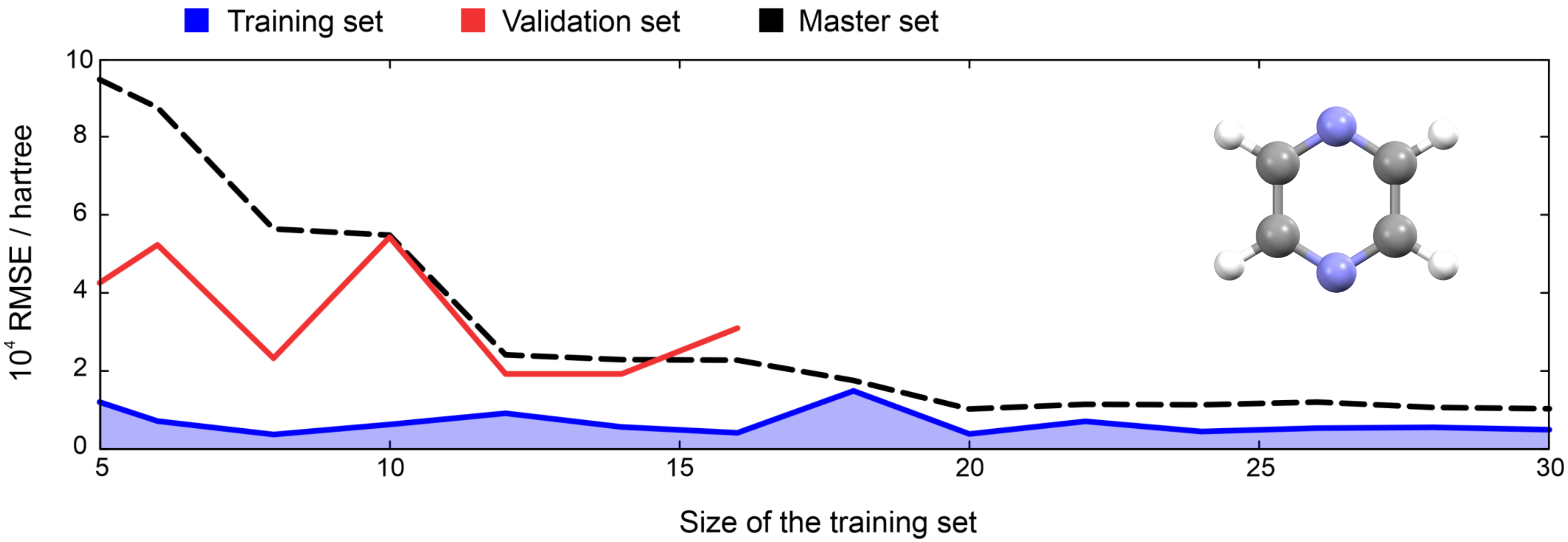


Figure 7. Deep reinforcement learning of *feed-forward* neural network with $1 \times (400)^8 \times 1$ architecture for modelling 1D cut of pyrazine's PES with respect to its 1st normal coordinate.

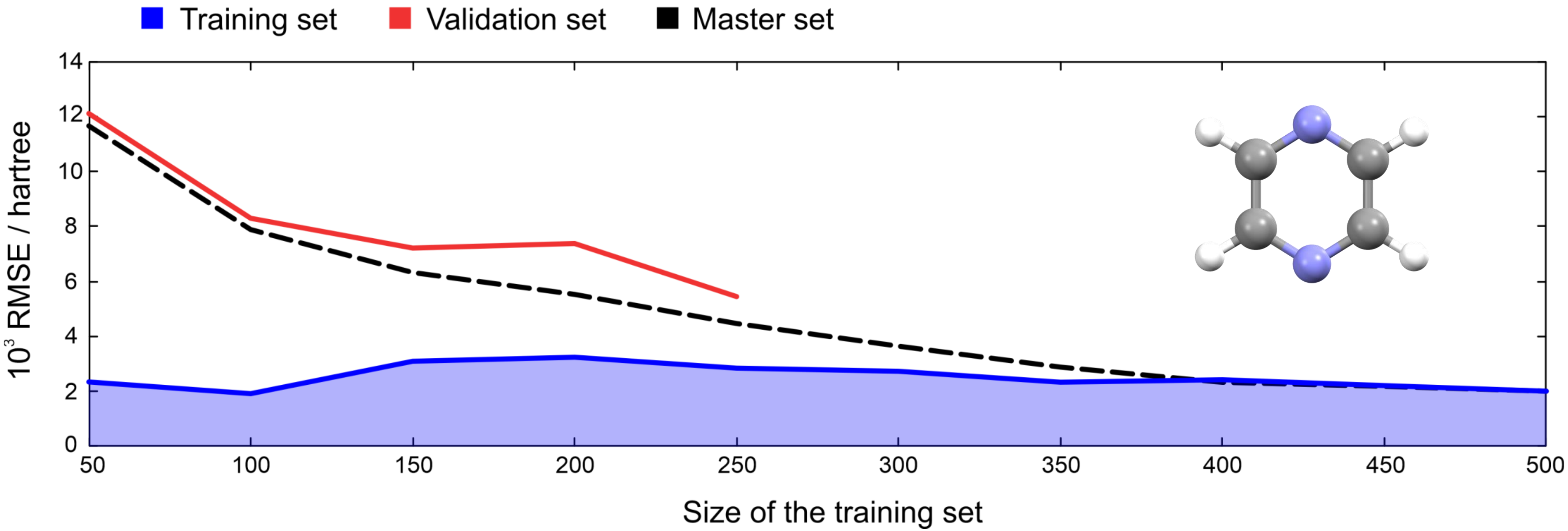


Figure 8. Deep reinforcement learning of *feed-forward* neural network with $30 \times (400)^8 \times 1$ architecture for modelling all 1D cuts of pyrazine's PES with respect to its normal coordinates.

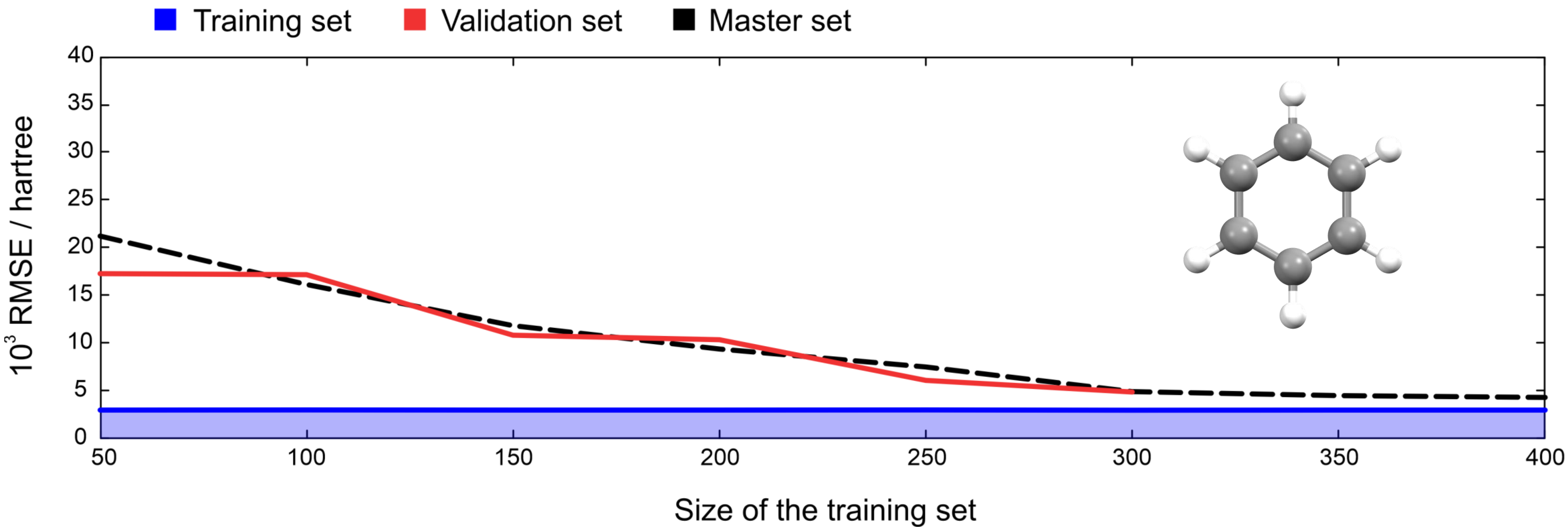


Figure 9. Deep reinforcement learning of *feed-forward* neural network with $36 \times (400)^8 \times 1$ architecture for modelling all 1D cuts of benzene's PES with respect to its normal coordinates.

Concluding remarks

- 1–4D cuts of benzene’s and pyrazine’s PES were successfully modeled using machine-learning MLR with polynomial expansion of their normal coordinates.

- Multi-layer neural networks have shown a great potential in modelling PESs and are currently under major development for various implementations in quantum chemistry.

- Further research is needed for *optimizing algorithms* concerning neural network architecture analysis and for *optimizing agent and policy* in deep reinforcement learning.

Thank you for your attention!